

PEMBUATAN APLIKASI *E-MAGAZINE* PENS-ITS PADA PLATFORM iOS

Ardianto Bayuaji Mahardika, Indra Adji Sulistijono, Akhmad Alimudin
Program Studi Teknologi Multimedia Broadcasting - Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember (ITS) Surabaya
Kampus PENS-ITS, Keputih, Sukolilo, Surabaya.
Telp : +62+031+5947280; Fax. +62+031+5946011
Email : black2punx@gmail.com

Abstrak – Perkembangan teknologi dewasa ini kian pesat, khususnya dalam bidang *touchscreen* dan *multitouch* yang membuat lebih interaktif sehingga diharapkan kedepannya majalah tidak hanya bisa dibaca namun juga dapat dilihat dan didengar.

Dalam proyek akhir ini akan dilakukan pembuatan e-magazine dari bentuk majalah menjadi aplikasi sistem informasi di dalam iPad. Aplikasi ini berisi tentang informasi mengenai kampus PENS ITS yang mencakup segala bidang, mulai dari informasi kampus PENS ITS secara umum seperti informasi jurusan, isi kampus maupun hasil karya yang diperoleh PENS selama ini.

Dalam pengerjaannya digunakan bahasa Cocoa dan Xcode sebagai *interface buildernya*. Dalam proyek akhir ini diharapkan mampu menciptakan sebuah aplikasi yang lebih efisien dan berinovasi tinggi khususnya dalam bidang teknologi informasi.

Kata kunci : App iPad , E-magazine, Cocoa Programming, Xcode

1. PENDAHULUAN

Dewasa ini perkembangan teknologi semakin pesat khususnya pada bidang *touchscreen* dan *multitouch*. Dengan adanya fitur tersebut aplikasi yang digunakan akan lebih interaktif dengan pembaca yang menggunakannya. Sedangkan *multitouch* sendiri lebih banyak digunakan pada layar *touchscreen* yang *capacitive* karena tidak menggunakan tekanan namun elektron pada jari.

Sehingga nantinya diharapkan aplikasi yang dibuat yaitu majalah digital dengan adanya fitur tersebut bisa lebih interaktif dengan pembaca. Diharapkan kedepannya majalah digital tersebut tidak hanya bisa dibaca namun juga dapat dilihat dan juga didengarkan.

1.1 PERUMUSAN MASALAH

Bagaimana membuat sebuah majalah digital pada platform iOS khususnya iPad

1.2 BATASAN MASALAH

Batasan masalah dari aplikasi proyek akhir ini adalah :

- Sistem Operasi untuk membuat MacOS
- Digunakan pada platform iOS
- Menggunakan bahasa pemrograman Cocoa
- Konten didapat dari newsletter PENS-ITS

1.3 TUJUAN

Tujuan dari pembuatan proyek akhir ini adalah membuat sebuah aplikasi yang interaktif dan efisien tempat dan waktu

2. TEORI PENUNJANG

2.1 Teknologi iOS

Lapisan lapisan yang ada pada iOS

Secara umum dalam iOS dan MacOS ada 4 layer penting yang harus diperhatikan karena dalam setiap layer mempunyai fungsi sendiri sendiri. Keempat layer tersebut adalah Cocoa Touch Layer, Media Layer, Core Service Layer dan Core OS layer. Perbedaan antara layer iOS dengan MacOS adalah pada layer paling atas. Sedangkan pada MacOS layer atas yaitu Cocoa Layer. Keempat layer tersebut mempunyai tugas sendiri sendiri yang mendukung satu sama lain. Untuk membuat suatu aplikasi pada iPhone OS, dibutuhkan sebuah komputer dengan OS Mac OS X yang mempunyai program aplikasi Xcode, Interface Builder, dan instrument.

Ketika akan membuat aplikasi pada iOS harus ditentukan terlebih dahulu pada layer Cocoa Touch interaksi apa yang akan digunakan pada user karena layer inilah yang berhubungan langsung dengan user seperti *multitouch*, *accelerate* dan *vibrate*. Setelah dari layer paling atas baru kemudian mulai layer bawahnya.

2.1.1 Cocoa Touch Layer



Gambar 2.3 Cocoa Touch

Sumber: iPhone Application Development.pdf Stanford.edu

Lapisan Cocoa Touch berisi kunci kerangka kerja untuk membangun aplikasi IOS. Lapisan ini mendefinisikan infrastruktur aplikasi dasar dan dukungan untuk teknologi utama seperti multitasking, input berbasis sentuhan, *push notification*, dan juga berbagai macam servis sistem tingkat tinggi. Hal pertama yang dilakukan ketika akan memulai mendesain aplikasi yang diinginkan maka yang harus diketahui adalah apakah layer ini memenuhi hal hal yang kita inginkan

2.1.2 Media Layer



Gambar 2.4 Media Layer

Sumber: iPhone Application Development.pdf Stanford.edu

Lapisan Media berisi grafis, audio, video dan teknologi diarahkan untuk memberikan sebuah penyajian multimedia terbaik yang tersedia pada perangkat mobile ini. Teknologi pada lapisan ini dirancang untuk memudahkan user untuk membangun aplikasi yang akan didengar dan dilihat oleh user

2.1.3 Core Service Layer

Layanan Core lapisan berisi layanan sistem dasar yang digunakan semua aplikasi. Bahkan jika Anda tidak menggunakan layanan ini

secara langsung, banyak bagian dari sistem dibangun di atas layer ini



Gambar 2.3 Core Service Layer

Sumber: iPhone Application Development.pdf Stanford.edu

2.1.4 Core OS Layer

Lapisan Core OS berisi beberapa fitur tingkat rendah yang sebagian besar teknologi lain yang dibangun di atas. Bahkan jika Anda tidak menggunakan teknologi tersebut secara langsung dalam aplikasi Anda, mereka kemungkinan besar digunakan oleh framework lain. Dan dalam situasi di mana Anda perlu secara eksplisit berurusan dengan keamanan atau berkomunikasi dengan hardware eksternal aksesori, Anda melakukannya dengan menggunakan kerangka dalam lapisan ini.



Gambar 2.3 Core Service Layer

Sumber: iPhone Application Development.pdf Stanford.edu

2.2 Xcode

Xcode adalah bagian dari apple development tools yang mendukung proyek manajemen, pengkodean, debugging, dan juga lainnya. Xcode merupakan Integrated Development Environment (IDE) yang memberikan semua tools yang diinginkan untuk mengatur dan membuat aplikasi pada iPhone, iPod Touch bahkan iPad.

2.3 Objective-C

Objective-C adalah bahasa pemrograman yang dikembangkan dari bahasa C digabung dengan gaya bahasa SmallTalk. Dengan kata lain, Objective-C pada OOP-nya bergaya SmallTalk yaitu menggunakan message passing sedangkan C++ menggunakan pemanggilan method. Objective-C saat ini banyak digunakan pada platform Mac OS X dan iOS (iOS adalah sistem operasi untuk iPhone, iPod Touch dan iPad. Dengan adanya framework Cocos2D yang notabene adalah framework untuk membuat game di iPhone, maka Objective-C makin banyak yang mempelajarinya. Catatan Cocos2D merupakan bagian Cocoa Touch API yang dibuat menggunakan Objective-C.

2.4 Aplikasi Pada iPhone/iPod Touch/iPad

Pada pembuatan aplikasi di iOS device sebelumnya harus mengerti tentang :

1. Pengetahuan dasar tentang Xcode dan Interface Builder
2. Bagaimana mendefinisikan suatu class baru dengan Obj-C class
3. Bagaimana mengatur memory termasuk membuat dan melepaskan suatu objek memory dalam Obj-C.
4. Fungsi dari delegate objects dalam pengaturan aplikasi.

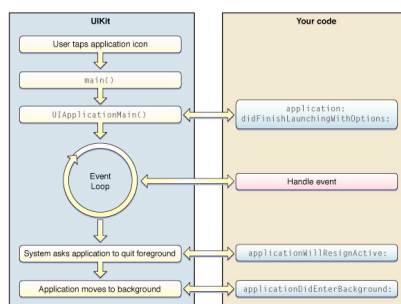
2.4.1 Aplikasi Core

Semua aplikasi iPhone menggunakan UIKit frameworks. UIKit menyediakan kunci objek yang dibutuhkan untuk menjalankan aplikasi dan untuk menangani suatu inputan dari pengguna dan menghasilkan isi gambar ke layar. Aplikasi core ini meliputi:

1. Aplikasi Arsitektur

A. Aplikasi Life Cycle

Aplikasi *Life Cycle* merupakan bagian dari *events* yang terjadi antara saat program mulai berjalan dan akhir dari aplikasi. Pada iOS ketika aplikasi dijalankan maka sistem akan melakukan transisi graphics dan memproses aplikasi itu dengan memanggil fungsi *main*.



Gambar 3.4 aplikasi Life Cycle.

Sumber:
<http://mobisynth.files.wordpress.com/2009/02/iphone-app-lifecycle1.png>

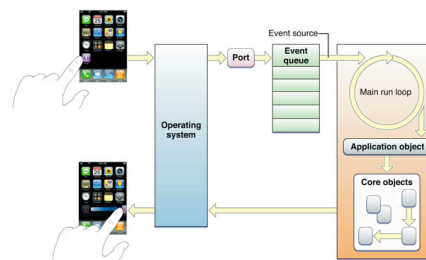
Ada beberapa sistemasi ataupun transisi aplikasi pada iOS yaitu :

- 1) Aplikasi yang tidak berjalan pada sistem
- 2) *Inactive*. Aplikasi yang berjalan di atas tapi yang tidak menerima event apapun. Aplikasi ini biasanya hanya akan memberitahu kepada user dengan mengirimkan sebuah alert. Aplikasi ini hanya tidak akan aktif untuk beberapa saat ketika user mengunci screen device atau ketika sistem memberikan peringatan untuk user agar merespon sebuah event yang ada seperti telepon yang masuk ataupun ketika menerima SMS.
- 3) *Active*. Aplikasi yang akan berjalan di foreground sistem dan juga menerima event
- 4) *Background*. Aplikasi yang akan berjalan di background sistem yang akan mengeksekusi kode

5) *Suspended* – Aplikasi yang akan berjalan di *background* sistem namun tidak mengeksekusi sebuah kode. Sistem akan mengubah aplikasi pada saat ini otomatis dan hanya pada waktu waktu tertentu

B. Event-Handling Cycle

Setelah fungsi `UIApplicationMain` menginisialisasi aplikasi, maka itu akan mulai mengerjakan infrastruktur yang dibutuhkan untuk mengatur *event* aplikasi dan *drawing cycle*. Event-Handling infrastruktur pada objek `UIApplication` mengambil setiap *event* lalu dijadikan *queue* dan menyampaikannya ke objek agar dapat ditangani. Pada iOS MultiTouch *event* modal fungsi sentuhan data merupakan ekapsulasi dalam single *event* (`UIEvent`)



Gambar 3.4 Event Handling Cycle.

Sumber:<http://mobisynth.files.wordpress.com/2009/02/event-loop2.png>

C. Fundamental Design Pattern

Desain pada UIKit frameworks tergabung dalam desain *pattern* yang ada pada aplikasi Cocoa di Mac OSX

Tabel 3.1 Design Pattern

Design Pattern	Deskripsi
Model-View-Controller (MVC)	MVC ini adalah cara untuk membuat hasil kode yang telah deprogram menjadi fungsional area yang bebas
Delegation	Delegation Design pattern merupakan cara agar dapat mengubah suatu objek yang kompleks tanpa harus mengubah sub class
Target Action	Kontrol menggunakan target-action untuk memberitahukan informasi kepada user. Ketika user berinteraksi dengan kontrol atau memulai untuk menjalankan program, kontrol itu akan mengirimkan suatu pesan yang berisi action ke objek yang dituju
Managed Memory Model	Mengatur memory yaitu dengan menggunakan bahasa Obj-C. Memory akan dihitung kemudian akan dijadikan referensi seiring dengan melepaskannya

Sumber: <http://howtomakeiphoneapps.com/home/2009/6/30/iphone-os-design-patterns.html>

2. Aplikasi Runtime Environment

a. Fast Launch, Low Use

Pada iOS hanya bagian depan aplikasi yang yang berjalan dalam satu waktu. Hal ini berarti bahwa aplikasi harus bekerja menginisialisasi dirinya sendiri agar dapat meminimalkan delay sekecil mungkin. *Runtime environment* didesain untuk kecepatan dan eksekusi program yang aman

b. Aplikasi SandBox

Untuk alasan pengamanan iOS membatasi aplikasi termasuk pengaturan dan data ke lokasi unique pada file system. Pembatasan itu merupakan bagian dari fitur pengamanan yang disebut aplikasi sandbox. Sandbox adalah suatu paket kontrol yang membatasi aplikasi mengakses file, pengaturan, jaringan dan juga perangkat hardware

c. Virtual Memory System

Mengatur program memory, iOS pada dasarnya menggunakan virtual memory yang sama pada sistem Mac OSX

d. Automatic Sleep Timer

Cara agar iOS menghemat baterai adalah dengan mengaktifkan *automatic sleep timer*. Jika sistem tidak menemukan sentuhan lewat *event* pada layar pada waktu yang lama, maka *device* akan mengurangi power lampu layar

3. Aplikasi Bundle

a. Information Property List

information property list adalah file yang bernama info.plist yang termasuk proyek iPhone yang dibuat oleh Xcode. Elemen elemen pada informasi list diatur dalam hirarki yang setiap node berisi array, dictionary, string atau scalar tipe lainnya. Gambar di bawah merupakan contoh file .plist yang terdapat dalam *final project* ini

b. Application icon and launch image

file untuk memperlihatkan icon image harus mempunyai nama "Icon.png" yang terletak di property CFBundleIconFile pada file "info.plist". Ukuran aplikasi icon harus maksimal 57x57 pixel. Sedangkan pada iPad maksimal 72x72 pixel dikarenakan resolusi warna juga berbeda.

c. Nib Files

Nib file adalah data file yang menyimpan paket "freeze-dried" objek. Aplikasi menggunakan nib file kebanyakan untuk menyimpan *windows* dan *views* yang didesain sebagai user interface. Interface builder adalah *environment* visual desain yang digunakan untuk membuat nib file.

4. Handling Critical Application Task

a. Initialization and termination

Selama inisialisasi dan terminasi, kelas UIApplication mengirim pesan ke delegate aplikasi untuk melakukan suatu tugas yang tidak perlu. Karena aplikasi pada iOS harus berakhir sebelum lainnya bisa berjalan, waktu yang diperlukan untuk mengeksekusi inisialisasi dan terminasi kode seharusnya menjadi sekecil mungkin

b. Responding on Interruptions

Ketika aplikasi mulai berakhir, sistem bisa menginterup aplikasi secara berkala yang bertujuan untuk memberikan

respon user ke *event* yang penting. Seperti contoh, aplikasi yang bisa diinterup oleh telepon yang datang, SMS, alarm, atau user menekan tombol *sleep* pada device. Jika user tidak menanggapi interupsi, maka aplikasi akan tetap berlanjut. Dan jika user memutuskan untuk menanggapi maka sistem akan mengakhiri program aplikasi yang berjalan tersebut.

- c. Observing Low-Memory Warning
iOS memberitahukan suatu informasi pada aplikasi itu ketika angka memory yang ada menurun di bawah yang seharusnya. Jika itu terjadi maka user harus mengkosongkan memory sebanyak-banyaknya dengan melepaskan objek yang tidak diinginkan atau membersihkan memory yang berjalan di *background*

3. METODOLOGI

1. Perancangan sistem

Dalam pembuatan majalah digital ini pertama-tama harus mempelajari literatur yang berkaitan dengan proyek ini. Hal-hal yang harus dipelajari, antara lain bahasa pemrograman Objective-C dan desain gambar menggunakan Photoshop melalui buku-buku maupun data-data yang didapatkan melalui internet. Setelah itu, kita akan mendesain tampilan aplikasi pada iPad. Dalam desain, terdapat beberapa menu yang mencakup semua informasi tentang PENS

2. Pembuatan dan Pengujian

Setelah melakukan perancangan sistem maka selanjutnya adalah membuat dan melakukan pengujian terhadap sistem tersebut

3. Integrasi Pengujian Sistem

Integrasi pengujian sistem guna mengetahui permasalahan-permasalahan yang mungkin muncul diantaranya lain pengujian :

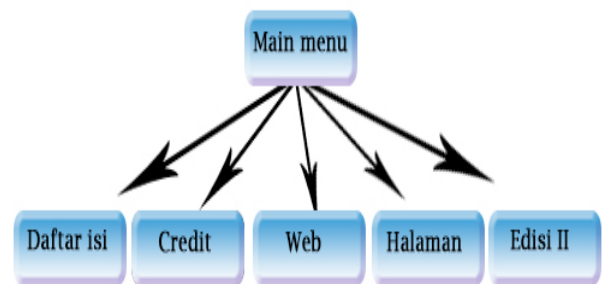
1. Keberhasilan dalam melakukan menggabungkan semua aspek dengan coding
2. Keberhasilan dalam menjalankan aplikasi ke dalam iPad dengan tanpa *bug*

4. Eksperimen dan analisa sistem

Dari hasil perancangan dilakukan realisasi / pembuatan perangkat lunak dan diadakan pengukuran atau pengujian masing-masing bagian dari perangkat lunak tersebut sebelum dilakukan integrasi. Sistem yang sudah dibangun, terintegrasi dan unjuk kerjanya bisa dijadikan sistem aplikasi majalah digital

dan digunakan juga untuk eksperimen aplikasi berbagai informasi tentang kampus PENS terhadap mahasiswa. Analisa hasil simulasi meliputi analisa rangkaian dan software dengan kerja sistem yang telah dikembangkan.

4. PERENCANAAN DAN IMPLEMENTASI SISTEM SECARA KESELURUHAN



Gambar 4.1. Blok Diagram Sistem Keseluruhan

Pada aplikasi ini terdapat 5 buah View yang masih masih terhubung satu sama lain. Main menu diatas adalah edisi pertama yang terhubung dengan menu daftar isi, menu untuk mengakses web, menu credit dan menu untuk edisi selanjutnya. Edisi selanjutnya mekanisme seperti pada main menu atau edisi pertama yaitu terhubung dengan semua menu view seperti pada gambar 1 blok diagram diatas. Pada menu daftar isi menggunakan object UITableView untuk setiap cell diisi dengan halaman yang dituju. Menu web menggunakan object UIWebView untuk menampilkan isi dari web EEPIS sehingga untuk melihat berita terbaru dari EEPIS dapat dilihat melalui menu ini. Menu credit menggunakan UIImageView yang me *load* gambar yang sudah dibuat sebelumnya. Dan menu edisi dua sendiri sama seperti menu edisi pertama namun isi konten yang berbeda

5. ANALISA DAN HASIL PENGUJIAN

5.1 Instalasi ke dalam *device*

untuk memasukkan atau menginstall aplikasi tersebut ke dalam *device* harus melalui beberapa tahap yang disebut provisioning. Tahapannya adalah sebagai berikut :

1. Membuat Provisioning Profile
File *profile* provisioning diperlukan untuk menginstruksikan ke xcode *device* apa yang

- akan digunakan nanti dan deskripsi tentang aplikasi yang dibuat ditulis disini
2. Download file Provisioning
Setelah membuat *profile* maka selanjutnya adalah mendownload file tersebut dan dimasukkan ke dalam xcode.
 3. Download Serifikat Developer
Sertifikat ini diperlukan untuk verifikasi apakah benar benar developer yang sudah teregistrasi ke dalam iOS Developer, karena untuk menginstall ke dalam *device* terlebih dahulu mendaftar ke dalam program tersebut.
 4. Running Xcode
Setelah semuanya selesai maka tahap terakhir adalah *merunning* proyek cocoa tersebut ke dalam iPad.

Hasil yang didapat setelah dijalankan di dalam iPad adalah aplikasi berjalan dengan normal sesuai dengan di simulator xcode, namun karena keterbatasan resource memori pada iPad maka ketika berganti ke edisi selanjutnya agak berat

6. KESIMPULAN

1. Sistem yang dibuat telah berhasil dengan baik, hal ini ditandai dengan ketika diinstall ke dalam *device* tidak ada *bug* atau aplikasi berjalan dengan normal
2. Ketika penulis mengaplikasikan pada iPad ternyata ada perbedaan dengan ketika menggunakan di simulator pada waktu menuju ke menu edisi dua, karena memori pada iPad sendiri dibatasi penggunaannya pada setiap aplikasi, maka ketika mengakses pertama kali ke edisi selanjutnya agak lambat



Gambar 5.1 Aplikasi yang dijalankan pada iPad



Gambar 5.2 Memutar Video